Research Article

Geometric Collocation Method on $SO(3)$ with Application to Optimal Attitude Control of a 3D Rotating Rigid Body

Xiaojia Xiang, Lizhen Wu, Lincheng Shen, and Jie Li

College of Mechatronics Engineering and Automation, National University of Defense Technology, Changsha, Hunan 410073, China

Correspondence should be addressed to Jie Li; leonlee2009@163.com

Received 22 August 2015; Accepted 13 October 2015

Academic Editor: Naohisa Otsuka

Copyright © 2015 Xiaojia Xiang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The collocation method is extended to the special orthogonal group $SO(3)$ with application to optimal attitude control (OAC) of a rigid body. A left-invariant rigid-body attitude dynamical model on $SO(3)$ is established. For the left invariance of the attitude configuration equation in body-fixed frame, a geometrically exact numerical method on $SO(3)$, referred to as the geometric collocation method, is proposed by deriving the equivalent Lie algebra equation in $\mathfrak{so}(3)$ of the left-invariant configuration equation. When compared with the general Gauss pseudo-spectral method, the explicit RKMK, and Lie group variational integrator having the same order and stepsize in numerical tests for evolving a free-floating rigid-body attitude dynamics, the proposed method is higher in accuracy, time performance, and structural conservativeness. In addition, the numerical method is applied to solve a constrained OAC problem on $SO(3)$. The optimal control problem is transcribed into a nonlinear programming problem, in which the equivalent Lie algebra equation is being considered as the defect constraints instead of the configuration equation. The transcription method is coordinate-free and does not need chart switching or special handling of singularities. More importantly, with the numerical advantage of the geometric collocation method, the proposed OAC method may generate satisfying convergence rate.

1. Introduction

Attitude control is one of the basic technologies that an object's motioning relies on, and thus it has been the research focus in motion control technology. For object motion, to enhance its performance and security, it is required to save its energy consumption and advance control performance itself, and the new control methods continue to emerge. There is no doubt that optimal control theory is the most natural framework toward conformity with meeting the above requirements. However, except that the linear quadratic issues such simple question, analytical solutions are seldom available or even possible. As a result, more often than not, one resorts to numerical techniques. Numerical methods for solving optimal control problems are divided into two major classes [1]: indirect methods and direct methods. In an indirect method, the calculus of variations [2] and the Pontryagin maximum principle [3] are used to determine the first-order optimality conditions of the original optimal control problem, so, it converts the optimal control problem

to a two-point boundary-value problem, including indirect shooting method [4] and indirect multiple-shooting method [5]. The primary advantage of the indirect methods lies in their high accuracy and assurances that the solution satisfies the necessary optimality conditions. However, indirect methods also have significant disadvantages. First, the necessary optimality conditions must be derived analytically; for most problems this derivation is difficult; Second, the radius of convergence is typically small. Third, a guess for the costates must be provided. In a direct method, the state and/or control of the optimal control problem is discretized in some manner and the problem is transcribed into a nonlinear programming problem (NLP), and, then, the NLP is solved using well-known optimization techniques, including direct shooting, collocation methods. Although not as high in precision as indirect methods, direct methods also have advantages because the optimality conditions do not need to be derived, and they have a large radius of convergence.

In general, the attitude of an object is represented by the Euler angle. However, when the object is regarded as a single

rigid body in three-dimensional space, the configuration of its rotation actually constructs a special Lie group called the special orthogonal group $SO(3)$ [6]. In recent years, some researchers have applied optimal control on a Lie group to attitude control of a 3D rotating rigid body, to reduce the computational burden, or to provide a coordinate-free approach to avoid expensive chart switching, special handling of singularities, and numerical drift [7, 8]. Bloch et al. [9] and Hussein et al. [10] study a discrete variational optimal control problem for the rigid body on $SO(3)$, and they describe the dynamical equations of the mechanical system as a geometric structure-preserving model referred to as a Lie group variational integrator and then use Lagrange's method to derive the discrete necessary conditions (i.e., two-point boundary-value problem) which are solved by a standard nonlinear root-finding method. Lee et al. [7] propose an accurate computational approach for a nonconvex optimal attitude control (OAC) for a rigid body, in which the use of geometrically exact computations on $SO(3)$ guarantees that the approach has excellent convergence properties. Similar to [7], the problem is also formulated directly as a discrete time optimization problem using a Lie group variational integrator; the discrete time necessary conditions for optimality are derived and solved by an efficient computational shooting method. Kobilarov and Marsden [11] extend the above conclusion to more general optimal control problems of mechanical systems on Lie groups and develop an efficient numerical method that exploits the structure of the state space and preserves the system motion invariants. The above-mentioned works have some common features: first, instead of discretizing equations of motion, they all use a discrete variational integrator obtained from the discrete Lagrange-d'Alembert principle, a process that better approximates the equations of motion; second, they are all indirect optimal control methods. In this paper, we turn to direct methods to solve the OAC problem on $SO(3)$. Among the direct methods, the collocation method, also known as a pseudo-spectral method, is widely used in rigid body attitude optimization (including UAV, satellite, space shuttle, launch vehicle, and supersonic aircraft) by virtue of its high accuracy and spectral (or exponential) convergence rate [1]. In the context of a Lie group, to our knowledge, Moulla et al. [12] are the first to propose the concept of the "geometric pseudo-spectral method." They suggest a polynomial pseudo-spectral method that preserves the geometric structure of port-Hamiltonian systems, phenomenological laws, and conservation laws without introducing any uncontrolled numerical dissipation. However, their method is designed only for port-Hamiltonian systems in a special structure, which implies that the Dirac structure, therefore, could not be directly extended to the systems on $SO(3)$. Thereby, how to apply the pseudo-spectral method to the OAC problem on $SO(3)$ and conserve its geometric properties is still an open question, which is also the major topic studied in this paper.

In this work, we study a constrained attitude control problem for the rigid body on $SO(3)$ based on pseudo-spectral collocation method. First, we establish a left-invariant rigid-body attitude dynamical model in body-fixed frame on $SO(3)$. Second, for the left invariance of attitude configuration

in the body-fixed frame, we provide a discretization mechanism known as the geometric collocation method on $SO(3)$, which may improve the accuracy of evolving trajectory and the convergence rate through applying different collocation strategies to the configuration equation and the velocity equation, respectively. For the configuration equation, since its geometric properties could not be preserved if the general pseudo-spectral method is directly applied to it, drawing on Engø's idea about the equivariant map [13], we transform the configuration equation on $SO(3)$ into an equivalent equation in the corresponding Lie algebra space [14] and then the solutions to the equivalent equation at discrete points are transformed into configuration at corresponding points via the coordinate map. For the velocity equation, we can use the general pseudo-spectral method directly to compute the velocities at the discrete points that are the same as those of configuration, since the velocity belongs to the Lie algebra space that is isomorphic to the Euclidean space. Under the condition that it has the same 4th order and large stepsize, the proposed method is compared with the general Gauss pseudo-spectral method [15], explicit RKMK [16], and Lie group variational integrator [7, 8]. We find that the accuracy of the proposed method is superior to other methods, except that the explicit RKMK demonstrates a higher convergence rate than the other methods. Furthermore, the method could preserve the structural characteristics of the Lie group precisely. Finally, we apply the above numerical method to a constrained OAC problem. In addition, the numerical method is used to solve the constrained OAC problem on $SO(3)$. The optimal control problem is transcribed into a nonlinear programming problem, in which the equivalent Lie algebra equation is considered as the defect constraints instead of configuration equation. The transcription method is coordinate-free and does not need chart switching or special handling of singularities. More importantly, with the numerical advantage of the geometric collocation method, the OAC method exhibits a high convergence rate.

The rest of this paper is organized as follows. Section 2 proposes a left-invariant rigid-body attitude dynamical model on $SO(3)$. The geometric collocation method for the left-invariant attitude dynamics system on $SO(3)$ is given in Section 3, where Section 3.1 provides a general pseudo-spectral collocation method in the Euclidean space, Section 3.2 presents the Legendre-Gauss pseudo-spectral collocation method on $SO(3)$, and, then, the corresponding algorithm is given in Section 3.3, and numerical tests are carried out on a free-floating rigid body model in comparison with four other numerical methods in Section 3.4. In Section 4, the proposed collocation is applied to a typical OAC problem and compared with several prior works. Finally, the conclusions and future work are outlined in Section 5.

2. Kinematics and Dynamics of Attitude Motion of a Rigid Body on $SO(3)$

2.1. Left-Invariant Mechanical System in a Body-Fixed Frame. The motion of a rigid body on Lie group G could be described in a space frame or a body-fixed frame. To avoid confusion,

we would consider the motion of a single rigid body in a body-fixed frame in the sequel.

Definition 1 (left-invariant mechanical systems on Lie group G [6]). In a body-fixed frame, the motion of a rigid body could be described as a left-invariant mechanical system as follows:

$$\dot{g} = T_e L_g \circ \hat{\xi} \in T_g G, \quad \forall g \in G, \quad (1)$$

where g denotes the configuration of a rigid body in G and $\hat{\xi} = T_{g^{-1}} \dot{g} \in T_e G$ denotes the velocity of the rigid body in the body-fixed frame, where $T_e G := \mathfrak{g}$, known as the Lie algebra, is the tangent space at the identity e of G and “ $\wedge$ ” is the hat operator satisfying isomorphism $\hat{\cdot} : \mathbb{R}^n \rightarrow \mathfrak{g}$ with “ $\vee$ ” being its inverse mapping. $L_g : G \rightarrow G$ represents the left translation; that is, $L_g(h) = gh, \forall h \in G$, and $T_e L_g$ is its tangent space at e . For the matrix group, such as $\text{SO}(3)$, $\text{SE}(2)$, and $\text{SE}(3)$, (1) is equivalent to $\dot{g} = g \cdot \hat{\xi}$. “Left-invariant” implies that the mechanical system is invariant under left multiplication by constant matrices.

2.2. Kinematics and Dynamics of Attitude Motion of a Rigid Body on $\text{SO}(3)$. $\text{SO}(3)$ is called the special orthogonal group, whose group element is the rotational component of the body-fixed frame $\{b\}$ relative to the space frame $\{a\}$ [6]. The left-invariant kinematics of attitude motion of a rigid body on $\text{SO}(3)$ are given by

$$\dot{R}_{ab}(t) = R_{ab}(t) \cdot \hat{\omega}_{ab}^b(t) := G(\omega_{ab}^b(t), R_{ab}(t)), \quad (2)$$

where $R_{ab}(t)$ and $\omega_{ab}^b(t)$ denote the attitude configuration and angular velocity of a rotating rigid body on $\text{SO}(3)$. The hat operator $\hat{\cdot} : \mathbb{R}^3 \rightarrow \mathfrak{so}(3)$, for all $y \in \mathbb{R}^3$, satisfying $\hat{\omega}_{ab}^b(t) \cdot y = \omega_{ab}^b(t) \times y$, where $\times$ is vector cross product [6].

And it is known that the attitude dynamics of the rigid body are generally satisfying [10, 11]

$$\begin{aligned} \dot{\omega}_{ab}^b(t) &= J^{-1} [J \omega_{ab}^b(t) \times \omega_{ab}^b(t) \\ &\quad + R_{ab}^*(t) \partial_{R_{ab}(t)} V(R_{ab}(t)) + T_{\text{ext}}(t)] \\ &:= F(\omega_{ab}^b(t), R_{ab}(t), T_{\text{ext}}(t)), \end{aligned} \quad (3)$$

where J is referred to as the inertia matrix of the rigid body, $R_{ab}^*(t) \partial_{R_{ab}(t)} V(R_{ab}(t))$ denotes the moment of the conservative force with $V(R_{ab}(t))$ being potential energy, and $T_{\text{ext}}(t) := B(t) \cdot u(t)$ is that of the nonconservative external force with $B(t)$ being the control conjugate vector.

3. Legendre-Gauss Geometric Pseudo-Spectral Collocation Method on $\text{SO}(3)$

3.1. Legendre-Gauss Pseudo-Spectral Collocation Method in the Euclidean Space. Before collocating the left-invariant mechanical system on $\text{SO}(3)$, we briefly describe the basic principle of the Legendre-Gauss pseudo-spectral collocation

(LG-PC) method [1]. Take, for example, solving the following differential equation in $\mathbb{R}^n$ at t_f ,

$$\begin{aligned} \dot{x}(t)|_t &= f(t, x(t)), \\ x(t_0) &= x_0 \in \mathbb{R}^n, \end{aligned} \quad (4)$$

$$t_0 \leq t \leq t_f.$$

First, we equally divide a time interval $[t_0, t_f]$ into several time subintervals $[t_k, t_{k+1}]$ with the stepsize being h and transform the time interval $t \in [t_k, t_{k+1}]$ into $\tau \in [-1, 1]$ via the affine transformation $t = [(t_{k+1} - t_k) \cdot \tau + (t_{k+1} + t_k)]/2$. Thus, accordingly, (4) over $[t_k, t_{k+1}]$ becomes

$$\dot{x}(\tau)|_\tau = \frac{t_{k+1} - t_k}{2} f(\tau, x(\tau)), \quad -1 \leq \tau \leq 1. \quad (5)$$

Let $\tau_1 < \dots < \tau_N$ be collocation points in $[-1, 1]$ and $\tau_0 = -1$. We approximate the solution of (5) by the following formula:

$$x(\tau) \approx X(\tau) = \sum_{i=0}^N x_i \cdot L_i(\tau), \quad (6)$$

where $L_i(\tau) = \prod_{j=0, j \neq i}^N (\tau - \tau_j) / (\tau_i - \tau_j)$ denotes Lagrange polynomials, satisfying the isolation property

$$L_i(c_j) = \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j. \end{cases} \quad (7)$$

Equation (6) together with the isolation property leads to $x_i = x(\tau_i)$, thereby $X(\tau_i) = x(\tau_i)$.

Through expressing the derivative of the Lagrange polynomials at the collocation points in differential matrix form $D = [d_{ji}] \in \mathbb{R}^{N \times (N+1)}$, where

$$\begin{aligned} d_{ji} &= \dot{L}_i(c_j) = \sum_{l=0}^N \frac{\prod_{k=0, k \neq i, l}^N (c_j - c_k)}{\prod_{k=0, k \neq i}^N (c_i - c_k)}, \\ j &= 1, \dots, N, \quad i = 0, \dots, N, \end{aligned} \quad (8)$$

we can write (5) at the collocation points as a set of differential algebra equations:

$$\begin{aligned} \sum_{i=0}^N d_{ji} \cdot X(c_i) &= \frac{t_{k+1} - t_k}{2} f(c_j, X(c_j)), \\ j &= 1, \dots, N. \end{aligned} \quad (9)$$

Based on the above equations, we establish the following defect equations,

$$\varsigma_j = \sum_{i=0}^N d_{ji} \cdot X(c_i) - \frac{t_{k+1} - t_k}{2} f(c_j, X(c_j)), \quad (10)$$

$$j = 1, \dots, N,$$

and apply iterative algorithms to the above equations so as to determine the approximation to $x(\tau)$ at the collocation

points. Finally, according to the following formula, we obtain the approximation to (4) at the endpoint t_{k+1} of time sub-interval $[t_k, t_{k+1}]$:

$$x(t_{k+1}) \approx X(t_k) + \frac{t_{k+1} - t_k}{2} \sum_{j=1}^N \bar{\omega}_j \cdot f(c_j, X(c_j)), \quad (11)$$

where $\bar{\omega}_j = \int_{-1}^1 L_j(t) dt$ is quadrature weight corresponding to the collocation points. This approximate solution will be the initial value of $x(t)$ over $[t_{k+1}, t_{k+2}]$ and so on.

Remark 2. According to different selection methods of collocation points, pseudo-spectral collocation methods can be divided into two categories, namely, standard and orthogonal methods. Collocation points in orthogonal collocation are those obtained from the roots of either Chebyshev polynomials $T_N(\tau)$ or Legendre polynomials $P_N(\tau)$ belonging to orthogonal polynomials [17]. The benefit of using orthogonal collocation over standard collocation is that the quadrature approximation to a definite integral is extremely accurate [1]. Furthermore, according to whether the endpoint is a collocation point, pseudo-spectral collocation methods fall into three general categories [18]: Gauss methods, where neither of the endpoints -1 or 1 is a collocation point; Radau methods, where at most one of the endpoints -1 or 1 is a collocation point; and Lobatto methods, where both the endpoints -1 and 1 are collocation points. In the rest of this paper, we will develop our geometric collocation method based on Legendre-Gauss points, since when Legendre-Gauss methods are used for transcribing the continuous optimal control problem into a discrete nonlinear programming problem, it does not suffer from a defect in the optimality conditions at the boundary points for the endpoints are not collocation points [15].

3.2. Legendre-Gauss Geometric Pseudo-Spectral Collocation Method on $SO(3)$. For attitude kinematics (referred to as the “configuration equation”), it is well-known that the solution of (2) stays on $SO(3)$ for all $t \geq t_0$. How to use the pseudo-spectral collocation method to solve (2), while maintaining some exact geometric properties (such as Lie group invariants) of the configuration equation under discretization of $R_{ab}(t)$, is the main problem to be solved in this subsection. However, $SO(3)$ is a nonlinear manifold, and using the pseudo-spectral collocation method directly to discretize the configuration could not preserve the geometric properties of its solution. Reference [13] indicated that any differential equation in the form of an infinitesimal generator on a homogeneous space (a manifold with a transitive Lie group action) is locally equivalent to a differential equation on a Lie algebra corresponding to the Lie group acting on the homogeneous space. Also, in the context of $SO(3)$, the Lie algebra $\mathfrak{so}(3)$ corresponding to $SO(3)$ is isomorphic to $\mathbb{R}^3$ [9–11]. For the above-mentioned reasons, the differential equation on $\mathfrak{so}(3)$ is the natural choice of space for pseudo-spectral discretization, like many classical Lie group numerical methods. For this purpose, we transform (3) into an equivalent equation on $\mathfrak{so}(3)$, then use the pseudo-spectral

collocation method to solve the equivalent equation, and finally map its solution back to configuration space. As mentioned earlier, since the Lie algebra space is isomorphic to the Euclid space to which the attitudes dynamics belong, and before discretizing the configuration equation, we directly use the general pseudo-spectral method to collocate the angular velocities at the discrete points.

3.2.1. Collocation of Angular Velocities at Discrete Points. For attitude dynamics, since its variable $\omega_{ab}^b(t)$ belongs to $\mathfrak{so}(3)$ that is isomorphic to $\mathbb{R}^3$, (3) can be collocated at Gauss collocation points directly by the Legendre-Gauss pseudo-spectral collocation method as follows:

$$\begin{aligned} \omega_{ab}^b(\tau_j) &= \frac{t_{k+1} - t_k}{2} \sum_{i=1}^N \bar{d}_{ji} \\ &\cdot F(\omega_{ab}^b(\tau_i), R_{ab}(\tau_i), T_{\text{ext}}(\tau_i)) \\ &- \left(\sum_{i=1}^N \bar{d}_{ji} \cdot d_{i0} \right) \cdot \omega_{ab}^b(t_k), \quad j = 1, \dots, N. \end{aligned} \quad (12)$$

Similarly, the approximate solutions of $\omega_{ab}^b(\tau_j)$, $j = 1, \dots, N$, are approached by several times of the Gauss-Seidel type iterative operation. Then, the approximation solutions are used to obtain the velocity at the endpoint t_{k+1} :

$$\begin{aligned} \omega_{ab}^b(t_{k+1}) &= \omega_{ab}^b(t_k) \\ &+ \frac{t_{k+1} - t_k}{2} \sum_{i=1}^N \bar{\omega}_i \cdot F(\omega_{ab}^b(\tau_i), R_{ab}(\tau_i), T_{\text{ext}}(\tau_i)). \end{aligned} \quad (13)$$

3.2.2. Collocation of Configurations at Discrete Points. To transform (2) on $SO(3)$ into an equivalent differential equation evolving on $\mathfrak{so}(3)$, we briefly describe the basic idea of the equivalent map.

Definition 3 (equivariant map [13]). Let $\mathcal{M}$ and $\mathcal{N}$ be manifolds and let G be a Lie group which acts on $\mathcal{M}$ by $\Phi_g : \mathcal{M} \rightarrow \mathcal{M}$ and on $\mathcal{N}$ by $\Psi_g : \mathcal{N} \rightarrow \mathcal{N}$. A smooth map $f : \mathcal{M} \rightarrow \mathcal{N}$ is called equivariant with respect to these actions if, for all $g \in G$,

$$f \circ \Phi_g = \Psi_g \circ f \quad (14)$$

which indicates the following diagram commutes.

Diagram commutes of the equivariant map are as follows:

$$\begin{array}{ccc} \mathcal{M} & \xrightarrow{f} & \mathcal{N} \\ \Phi_g \uparrow & (14) & \uparrow \Psi_g \\ \mathcal{M} & \xrightarrow{f} & \mathcal{N} \end{array} \quad (15)$$

First, from the definition of an action of G on $\mathcal{M}$, we can obtain an equivariant map $\Phi_g : G \rightarrow \mathcal{M}$ with respect to the

left translation action L_g of G on itself and an action G of Φ_g on $\mathcal{M}$:

$$\Phi_y \circ L_g = \Phi_g \circ \Phi_y. \quad (16)$$

It is known that there is a local coordinate map $f : \mathfrak{g} \rightarrow G$ on G with the Lie algebra $\mathfrak{g}$, and the most typical one is the exponential map $\exp : \mathfrak{g} \rightarrow G$. At this point, we need to find an action B_g of G on $\mathfrak{g}$ such that f will be an equivariant map with B_g and the left action of G on itself, namely,

$$f \circ B_g = L_g \circ f. \quad (17)$$

In case f is the exponential map, B_g is nothing other than the well-known Baker-Campbell-Hausdorff (BCH) formula $B_g(u) = \log(g \cdot \exp(u))$, where $\log : G \rightarrow \mathfrak{g}$ is called the logarithm map. Since the composition of two equivariant maps is also an equivariant map, we can construct an equivariant map $\Phi_y \circ f$ from $\mathfrak{g}$ to $\mathcal{M}$ with respect to the action B_g on $\mathfrak{g}$ and Φ on $\mathcal{M}$.

Diagram commutes of composition $\Phi_y \circ f$ are as follows:

$$\begin{array}{ccccc} \mathfrak{g} & \xrightarrow{f} & G & \xrightarrow{\Phi_y} & \mathcal{M} \\ B_{\exp(t\xi)} \uparrow & (17) \uparrow & L_g & (16) \uparrow & \Phi_{\exp(t\xi)} \\ \mathfrak{g} & \xrightarrow{f} & G & \xrightarrow{\Phi_y} & \mathcal{M} \end{array} \quad (18)$$

Theorem 3.6 in literature [13] states that if ϕ is an equivariant map, then the infinitesimal generators of the action with respect to the same element $\xi \in \mathfrak{g}$ are ϕ -related vector fields. Thus, the infinitesimal generators ξ_g and $\xi_{\mathcal{M}}$ of the flows $B_{\exp(t\xi)}$ and $\Phi_{\exp(t\xi)}$, respectively, on $\mathfrak{g}$ and $\mathcal{M}$ are $\Phi_y \circ f$ -related; that is,

$$\xi_{\mathcal{M}} \circ \Phi_y \circ f = T\Phi_y \circ Tf \circ \xi_g. \quad (19)$$

Finally, we need to determine what ξ_g is, and the following theorem gives the conditions that it needs to meet.

Lemma 4 (see [13]). *Let $f : \mathfrak{g} \rightarrow G$ be a coordinate map on G and $\Phi_y \circ f$ equivariant with respect to the flows $B_{\exp(t\xi)}$ and $\Phi_{\exp(t\xi)}$. The infinitesimal generator of B_g satisfying (19) is $\xi_g(u) = df_u^{-1}(v)$. $df : \mathfrak{g} \rightarrow \mathfrak{g}$ is the trivialization Tf defined as $df_u = TR_{f(u)^{-1}} \circ Tf_u$.*

Based on Lemma 4 and other previous conclusions, we can derive the equivalent equation on $\mathfrak{so}(3)$ of the left-invariant configuration equation.

Proposition 5 (equivalent equation on $\mathfrak{so}(3)$ of the left-invariant configuration equation). *The configuration equation (2) has an equivalent equation on $\mathfrak{so}(3)$ as follows:*

$$\dot{u}(t) = df_{-u(t)}^{-1}(\omega_{ab}^b(t)) := \tilde{G}(\omega_{ab}^b(t), u(t)), \quad (20)$$

$$t_k \leq t \leq t_{k+1},$$

where $f : \mathfrak{so}(3) \rightarrow SO(3)$ is a local coordinate map on $SO(3)$ with df being its differential.

Proof. Without any loss of generality, we assume that the solution $R_{ab}(t)$ of (2) can be written in the following form:

$$R_{ab}(t) = R_{ab}(t_0) \cdot f(u(t)). \quad (21)$$

Then, by differentiating it [20], we can obtain

$$\begin{aligned} \frac{d}{dt} R_{ab}(t) &= \frac{d}{dt} (R_{ab}(t_0) \cdot f(u(t))) \\ &= R_{ab}(t_0) \cdot dL_{f(u(t))} \circ df_{-u(t)}(\dot{u}(t)) \\ &= R_{ab}(t_0) \cdot f(u(t)) \cdot df_{-u(t)}(\dot{u}(t)). \end{aligned} \quad (22)$$

Comparing the preceding formula with (2) and (21), we have

$$\omega_{ab}^b(t) = df_{-u(t)}(\dot{u}(t)). \quad (23)$$

Taking the inverse of the preceding formula, we obtain the equivalent equation (20) on $\mathfrak{so}(3)$ of the left-invariant configuration equation (2). $\square$

There are several choices of the coordinate map f , such as the exponential map and Cayley map [21]. In case f is the exponential map “exp” the vector field $df_{-u(t)}^{-1}$ on $\mathfrak{so}(3)$ can be represented as [20]

$$\begin{aligned} d\exp_{-u(t)}^{-1}(\omega_{ab}^b(t)) &= \omega_{ab}^b(t) + \frac{1}{2} \text{ad}_{u(t)}(\omega_{ab}^b(t)) \\ &\quad + \sum_{k=2}^{\infty} \frac{B_k}{k!} \text{ad}_{u(t)}^k(\omega_{ab}^b(t)), \end{aligned} \quad (24)$$

where the adjoint operator $\text{ad}_u : \mathfrak{g} \rightarrow \mathfrak{g}$ is a Lie bracket or commutator, $\text{ad}_u(\omega_{ab}^b(t)) = [u, \omega_{ab}^b(t)]$, satisfying the recursive expression $\text{ad}_u^k(\omega_{ab}^b(t)) = [u, \text{ad}_u^{k-1}(\omega_{ab}^b(t))]$, and B_k are Bernoulli numbers [21]:

$$B_k = \begin{cases} \frac{1}{6}, -\frac{1}{30}, \frac{1}{42}, -\frac{1}{30}, \frac{5}{66}, -\frac{691}{2730}, \dots & k = 2N, N \in \mathbb{Z}^+ \\ 0, & k = 2N+1, N \in \mathbb{Z}^+. \end{cases} \quad (25)$$

Now, we solve (20) by the LG-PC method as follows:

$$\begin{aligned} u(\tau_j) &= \frac{t_{k+1} - t_k}{2} \sum_{i=1}^N \tilde{d}_{ji} \cdot \tilde{G}(\omega_{ab}^b(\tau_i), u_{ab}(\tau_i)) \\ &\quad - \left(\sum_{i=1}^N \tilde{d}_{ji} \cdot d_{i0} \right) \cdot u(t_k), \quad j = 1, \dots, N. \end{aligned} \quad (26)$$

In view of the mechanical system being autonomous, we may use Gauss-Seidal type iteration to update the approximate solution of (26) as soon as possible [22]. As mentioned earlier, after obtaining the approximate solution, the velocity at the endpoint t_{k+1} can be obtained according to the following formula:

$$\begin{aligned} u(t_{k+1}) &= u(t_k) \\ &\quad + \frac{t_{k+1} - t_k}{2} \sum_{i=1}^N \tilde{\omega}_i \cdot \tilde{G}(\omega_{ab}^b(\tau_i), u_{ab}(\tau_i)). \end{aligned} \quad (27)$$

It is worth noting that, different from the LG-PC method in the Euclidean space, the initial value $u(t_k)$ in $[t_k, t_{k+1}]$ is not equal to the endpoint value $u(t_k)$ in $[t_{k-1}, t_k]$. It should be calculated through the logarithmic map $\log : \text{SO}(3) \rightarrow \mathfrak{so}(3)$:

$$u(t_k) = \log_{\text{SO}(3)} \left(R_{ab}^{-1}(t_k) \cdot R_{ab}(t_k) \right), \quad k = 0, 1, 2, \dots \quad (28)$$

with

$$\log_{\text{SO}(3)}(R) = \frac{\phi}{2 \sin \phi} (R - R^T) \in \mathfrak{so}(3), \quad (29)$$

where ϕ satisfies $\cos \phi = (\text{tr}(R) - 1)/2$, $|\phi| < \pi$ and $\text{tr}(R) \neq -1$. It is easy to know that $u(t_k)$, $k = 0, 1, 2, \dots$, are identically equal to zero.

At last, the attitude configuration $R_{ab}(t_{k+1})$ at the endpoint t_{k+1} can be reconstructed from $u(t_{k+1})$ according to $R_{ab}(t_{k+1}) = R(t_k) \cdot f(u(t_{k+1}))$, where $k \in \mathbb{Z}^+$ and the Cayley map should be selected as the coordinate map f , for two reasons [11]: first, it provides only an approximation to the integral curve defined by the exponential map, which is however easy to calculate; second, it does not involve trigonometric functions so that its derivatives do not have any singularities and thus will not cause any difficulties when used by gradient-based methods for solving NLP problems. The Cayley map satisfies the following formula:

$$\begin{aligned} \text{cay}_{\text{SO}(3)}(\omega_{ab}^b(t)) \\ = I_3 + \frac{4}{4 + \|\omega_{ab}^b(t)\|^2} \left[\hat{\omega}_{ab}^b(t) + \frac{(\hat{\omega}_{ab}^b(t))^2}{2} \right], \end{aligned} \quad (30)$$

where $\|\cdot\|$ is a standard Euclidean norm.

3.3. Algorithm of the Legendre-Gauss Geometric Pseudo-Spectral Collocation Method on SO(3) with Application to Attitude Motion of a Rigid Body. In accordance with the aforementioned LG-GPC method, we develop a Legendre-Gauss geometric pseudo-spectral collocation algorithm for integrating the kinematics and dynamics of attitude motion of a rigid body on SO(3). This algorithm would be further used to discretize the OAC problem in Section 4.

Algorithm 6 (LG-GPC). Inputs include time interval $[t_0, t_f]$, stepsize h , number of LG points N , initial attitude configuration $R_{ab}(t_0)$, initial angular velocity $\omega_{ab}^b(t_0)$, iterations of inner while loop num, and threshold of equivalent configuration (or velocity) deviation err.

Step 1. Initialization is as follows: let k be zero.

Step 2. Compute initial value of the equivalent configuration at t_0 in terms of $R_{ab}(t_0)$:

$$u(t_0) = \log_{\text{SO}(3)} \left(R_{ab}^{-1}(t_0) \cdot R_{ab}(t_0) \right) \equiv 0. \quad (31)$$

Step 3 (while $(t_k < t_f)$).

Step 3.1. Compute the LG points $\{\tau_1, \dots, \tau_N\}$ in $[t_k, t_{k+1}]$, differentiation matrix $D_{N \times (N+1)}$, and quadrature weights $\{\omega_1, \dots, \omega_N\}$.

Step 3.2. Set the current iterative step number $n = 0$ and configuration (or velocity) deviation $e = 1$.

Step 3.3. Let $\omega_{ab}^{b(n)}(\tau_i)$, $i = 1, \dots, N$, be $\omega_{ab}^b(t_k)$ let and $u^{(n)}(\tau_i)$, $i = 1, \dots, N$ be $u(t_k)$.

Step 3.4 (while $(n < \text{num}$ and $e > \text{err}$)).

Step 3.4.1. Obtain the submatrix $\tilde{D}_{N \times N}$ of the matrix $D_{N \times (N+1)}$:

$$\tilde{D} = D(:, 2 : N + 1). \quad (32)$$

Step 3.4.2. According to the following equations, obtain $[\dot{\omega}_{ab}^{b(n)}(\tau_1), \dots, \dot{\omega}_{ab}^{b(n)}(\tau_N)]^T$:

$$\begin{aligned} & \begin{bmatrix} \dot{\omega}_{ab}^{b(n)}(\tau_1) \\ \vdots \\ \dot{\omega}_{ab}^{b(n)}(\tau_N) \end{bmatrix} \\ &= \frac{t_{k+1} - t_k}{2} \begin{bmatrix} F(\omega_{ab}^{b(n)}(\tau_1), R_{ab}^{(n)}(\tau_1), T_{\text{ext}}(\tau_1)) \\ \vdots \\ F(\omega_{ab}^{b(n)}(\tau_N), R_{ab}^{(n)}(\tau_N), T_{\text{ext}}(\tau_N)) \end{bmatrix} \\ &- \begin{bmatrix} d_{01} \\ \vdots \\ d_{0N} \end{bmatrix} \omega_{ab}^b(t_k). \end{aligned} \quad (33)$$

Step 3.4.3. Update angular velocities $[\omega_{ab}^{b(n+1)}(\tau_1), \dots, \omega_{ab}^{b(n+1)}(\tau_N)]^T$ at LG points:

$$\begin{bmatrix} \omega_{ab}^{b(n+1)}(\tau_1) \\ \vdots \\ \omega_{ab}^{b(n+1)}(\tau_N) \end{bmatrix} = \tilde{D}^{-1} \cdot \begin{bmatrix} \dot{\omega}_{ab}^{b(n)}(\tau_1) \\ \vdots \\ \dot{\omega}_{ab}^{b(n)}(\tau_N) \end{bmatrix}. \quad (34)$$

Step 3.4.4. According to the following equations, obtain $[\dot{u}^{(n)}(\tau_1), \dots, \dot{u}^{(n)}(\tau_N)]^T$:

$$\begin{aligned} & \begin{bmatrix} \dot{u}^{(n)}(\tau_1) \\ \vdots \\ \dot{u}^{(n)}(\tau_N) \end{bmatrix} \\ &= \frac{t_{k+1} - t_k}{2} \begin{bmatrix} \widetilde{G}(\omega_{ab}^{b(n)}(\tau_1), u^{(n)}(\tau_1)) \\ \vdots \\ \widetilde{G}(\omega_{ab}^{b(n)}(\tau_N), u^{(n)}(\tau_N)) \end{bmatrix} \\ & - \begin{bmatrix} d_{01} \\ \vdots \\ d_{0N} \end{bmatrix} u(t_k). \end{aligned} \quad (35)$$

Step 3.4.5. Update the equivalent configuration $[u^{(n+1)}(\tau_1), \dots, u^{(n+1)}(\tau_N)]^T$ at LG points:

$$\begin{bmatrix} u^{(n+1)}(\tau_1) \\ \vdots \\ u^{(n+1)}(\tau_N) \end{bmatrix} = \widetilde{D}^{-1} \cdot \begin{bmatrix} \dot{u}^{(n)}(\tau_1) \\ \vdots \\ \dot{u}^{(n)}(\tau_N) \end{bmatrix}. \quad (36)$$

Step 3.4.6. Update the configuration $[R_{ab}^{(n+1)}(\tau_1), \dots, R_{ab}^{(n+1)}(\tau_N)]^T$ at LG points:

$$\begin{bmatrix} R_{ab}^{(n+1)}(\tau_1) \\ \vdots \\ R_{ab}^{(n+1)}(\tau_N) \end{bmatrix} = \begin{bmatrix} R_{ab}(t_k) \cdot \text{cay}(u^{(n+1)}(\tau_1)) \\ \vdots \\ R_{ab}(t_k) \cdot \text{cay}(u^{(n+1)}(\tau_N)) \end{bmatrix}. \quad (37)$$

Step 3.4.7. Compute the equivalent configuration deviation and the velocity deviation within two adjacent iterative steps, and, then, let the larger one be the final deviation e at the current step:

$$\begin{aligned} e = \max & \left\{ \text{abs} \left(\left\| \begin{bmatrix} \det(R_{ab}^{(n+1)}(\tau_1)^T R_{ab}^{(n)}(\tau_1)) \\ \vdots \\ \det(R_{ab}^{(n+1)}(\tau_N)^T R_{ab}^{(n)}(\tau_N)) \end{bmatrix} \right\|_{\infty} \right. \right. \\ & \left. \left. - 1 \right), \left\| \begin{bmatrix} \omega_{ab}^{b(n+1)}(\tau_1) - \omega_{ab}^{b(n)}(\tau_1) \\ \vdots \\ \omega_{ab}^{b(n+1)}(\tau_N) - \omega_{ab}^{b(n)}(\tau_N) \end{bmatrix} \right\|_{\infty} \right\}. \end{aligned} \quad (38)$$

Step 3.4.8. Substitute $\omega_{ab}^{b(n+1)}(\tau_i)$, $u^{(n+1)}(\tau_i)$, and $R_{ab}^{(n+1)}(\tau_i)$ into $\omega_{ab}^{b(n)}(\tau_i)$, $u^{(n)}(\tau_i)$, and $R_{ab}^{(n)}(\tau_i)$, respectively, where $i = 1, \dots, N$, and let $n = n + 1$.

Step 3.5. End while.

Step 3.6. Relying on the final iterative values $\omega_{ab}^{b(n)}(\tau_i)$, $R_{ab}^{(n)}(\tau_i)$ with $T_{\text{ext}}(\tau_i)$, $i = 1, \dots, N$ in inner while loop, compute the angular velocity $\omega_{ab}^b(t_{k+1})$ at the endpoint t_{k+1} :

$$\begin{aligned} \omega_{ab}^b(t_{k+1}) &= \omega_{ab}^b(t_k) + \frac{t_{k+1} - t_k}{2} \\ & \cdot \sum_{i=1}^N \omega_i \cdot F(\omega_{ab}^{b(n)}(\tau_i), R_{ab}^{(n)}(\tau_i), T_{\text{ext}}(\tau_i)). \end{aligned} \quad (39)$$

Step 3.7. Also, based on the final iterative values $\omega_{ab}^{b(n)}(\tau_i)$, $u^{(n)}(\tau_i)$, $i = 1, \dots, N$ in the inner while loop, compute the equivalent configuration $u(t_{k+1})$ at the endpoint t_{k+1} :

$$\begin{aligned} u(t_{k+1}) &= u(t_k) \\ & + \frac{t_{k+1} - t_k}{2} \sum_{i=1}^N \omega_i \cdot \widetilde{G}(\omega_{ab}^{b(n)}(\tau_i), u^{(n)}(\tau_i)). \end{aligned} \quad (40)$$

Step 3.8. Based on the previous formula, compute the configuration $R_{ab}(t_{k+1})$ at the endpoint t_{k+1} :

$$R_{ab}(t_{k+1}) = R_{ab}(t_k) \cdot \text{cay}(u(t_{k+1})). \quad (41)$$

Step 3.9. Let $t_k = t_{k+1}$, $k = k + 1$, and $u(t_k) = 0$.

Step 4. End while.

Remark 7. The LG points τ_i , $i = 1, \dots, N$, are the roots of the Legendre polynomial $P_N(\tau) = d^N[(\tau^2 - 1)^N]/[(2^N N!) d\tau^N]$ of degree N . In fact, it is convenient to compute the LG points via the eigenvalues of the following tridiagonal Jacobi matrix:

$$J = \begin{bmatrix} 0 & \beta_1 & & & \\ \beta_1 & 0 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 0 & \beta_{N-1} \\ & & & \beta_{N-1} & 0 \end{bmatrix}, \quad \beta_n = \frac{n}{\sqrt{4n^2 - 1}}. \quad (42)$$

The LG pseudo-spectral differentiation matrix $D = [d_{ji}] \in \mathbb{R}^{N \times (N+1)}$ is given as [22]

$$d_{ji} = \begin{cases} \frac{(1 + \tau_j) \dot{P}_N(\tau_j) + P_N(\tau_j)}{(\tau_j - \tau_i) [(1 + \tau_i) \dot{P}_N(\tau_i) + P_N(\tau_i)]}, & i \neq j \\ \frac{(1 + \tau_i) \ddot{P}_N(\tau_i) + 2\dot{P}_N(\tau_i)}{2[(1 + \tau_i) \dot{P}_N(\tau_i) + P_N(\tau_i)]}, & i = j \end{cases} \quad (43)$$

FIGURE 1: Accuracy and computational efficiency of LG-GPC when compared with LG-PC and RKMK.

with

$$\dot{P}_N = \frac{-(N+1)P_N(\tau)}{1-\tau^2}. \quad (44)$$

The LG weights corresponding to the LG points are computed as

$$\omega_i = \frac{2}{(1-\tau_i^2)\dot{P}_N^2(\tau_i)}. \quad (45)$$

Remark 8. In Step 3.3, both $u^{(\text{step})}(\tau_1)$ and $u^{(\text{step})}(\tau_2)$ cannot be set to $u(t_k)$, because $u(t_k)$ shows the projection of distance between $R_{ab}(t_{k-1})$ and $R_{ab}(t_k)$ on the tangent space.

3.4. Numerical Integration Tests of the LG-GPC Method. In this subsection, we will test the validity and efficiency of the proposed method through numerical integration of the following free-floating rigid body kinematics (46) and dynamics (47) on $\text{SO}(3)$. Note that (46) and (47) are equivalent to (6) and (7) under the condition of zero external force. Table 1 summarizes the parameters used by numerical tests. Consider

$$\dot{R}_{ab}(t) = R_{ab}(t) \cdot \hat{\omega}_{ab}^b(t) := G(\omega_{ab}^b(t), R_{ab}(t)), \quad (46)$$

$$\begin{aligned} \dot{\omega}_{ab}^b(t) &= J^{-1} [J\omega_{ab}^b(t) \times \omega_{ab}^b(t)] \\ &:= F(\omega_{ab}^b(t), R_{ab}(t)). \end{aligned} \quad (47)$$

To ensure that numerical integration test is more comparable, we consider the 4th-order LG-GPC method and select three other 4th-order numerical methods for comparison, including ode45 [23], explicit 4th-order RKMK [16], and 4th-order LG-PC method [22]. Furthermore, we compare the proposed numerical method with the Lie group variational

TABLE 1: Parameters of numerical integration tests [19].

Physical quantity	Symbol	Value
Initial time	t_0	0 s
Terminal time	t_f	240 s
Moment of inertia	J	$\text{diag}[1, 2.8, 2] \text{ kg}\cdot\text{m}^2$
Initial configuration	$R_{ab}(t_0)$	$I_{3 \times 3}$
Initial angular velocity	$\omega_{ab}^b(t_0)$	$[0.5, -0.5, 0.4]^T \text{ rad/s}$

integrator (LGVI) which is also based on geometrically exact computations on $\text{SO}(3)$ [19]. Among the previous methods, ode45 is the built-in integrator of MATLAB, which is a variable-step method and whose results would be used as the ground truth with a specified tolerance 10^{-14} . RKMK is a Lie group method [16]. Note that [24] states that the interpolatory quadrature formula with N Gauss points is of $2N$ th order. To this end, both the 4th-order LG-GPC method and the LG-PC method depend on 2 LG points. In addition, in the LG-GPC and LG-PC methods, the iterations of the inner while loop num and the threshold of the equivalent configuration (or velocity) deviation err are set to be 30 and 10^{-14} , respectively. Since LGVI has an implicit equation, we need to solve it by using the Newton iterative approach, where the termination error of iteration and the maximum estimated value of the function are set to be $1e^{-14}$ and 9000, respectively. Numerical integration tests are all carried out in the same stepsize.

Figure 1 gives the angular configuration deviation and the average runtime per update of LG-GPC method when compared with LG-PC, RKMK, and LGVI, respectively, at five different steps. Figure 1(a) shows the angular configuration deviation against steps by calculating $\|\log_{\text{SO}(3)}(R_{\text{ode45}}^T R_{ab})\|_2$, where R_{ode45} denotes the angular configuration of the free-floating rigid body which is computed by ode45. It has been

FIGURE 2: Lie group structural deviation of LG-GPC when compared with LG-PC, RKMK, and LGVI.

illustrated that the accuracy of LG-GPC is superior to those of LG-PC, RKMK, and LGVI. Figure 1(b) shows the average runtime per update of LG-GPC methods when compared with LG-PC, RKMK, and LGVI. As it can be seen, the computational efficiency of LG-GPC is inferior to that of RKMK, because RKMK is an explicit method. However, the proposed method requires less computation time than the classical LG-PC method and is nearly 1.5 times as efficient as the LG-PC method on average. More obviously, the computational efficiency of the proposed method is superior to that of LGVI, since the latter needs the Newton iteration approach to solve the implicit approach.

Additionally, the Lie group structural conservativeness of LG-GPC is compared with those of LG-PC, RKMK, and LGVI. Considering that the angular configuration is parameterized by unit quaternion in the ode45 method, we only compare the proposed method with the LG-PC method, explicit RKMK, and LGVI. As mentioned above, since the group element on $SO(3)$ satisfies $R_{ab}R_{ab}^T = I$, we compute the infinite norm of $R_{ab}R_{ab}^T - I_{3 \times 3}$ for evaluating the structural conservativeness [19]. Figure 2 illustrates that the LG-PC method belonging to the Euclidean space has no group structural conservativeness, because it is not a Lie group method. However, RKMK, LGVI, and LG-GPC method on the Lie group are able to preserve the Lie group structure with an accuracy approaching the machine accuracy 10^{-15} , and what is more, the LG-GPC is even superior to RKMK and LGVI.

4. OAC on $SO(3)$ Based on LG-GPC Discretization

4.1. Description of the OAC Problem on $SO(3)$. Consider the following OAC problem with differential kinematical and dynamical constraints on $SO(3)$.

Problem 9 (OAC problem with kinematical and dynamical constraints on $SO(3)$). Minimize the continuous-time cost

$$\mathcal{J} = \frac{1}{2} \int_{t_0}^{t_f} \langle u(t), Pu(t) \rangle_{\mathbb{R}^m} dt, \quad (48)$$

where $u(t) \in \mathbb{R}^m$ is the control vector and $P \in \mathbb{R}^{m \times m}$ is the weighting matrix, where $m \leq 3$,

(1) satisfying the differential kinematical and dynamical constraints on $SO(3)$,

$$\begin{aligned} \dot{R}_{ab}(t) &= R_{ab}(t) \cdot \hat{\omega}_{ab}^b(t) := G(\omega_{ab}^b(t), R_{ab}(t)), \\ \dot{\omega}_{ab}^b(t) &= J^{-1} [J\omega_{ab}^b(t) \times \omega_{ab}^b(t) \\ &\quad + R_{ab}^*(t) \partial_{R_{ab}(t)} V(R_{ab}(t)) + T_{\text{ext}}(t)] \\ &:= F(\omega_{ab}^b(t), R_{ab}(t), T_{\text{ext}}(t)) \end{aligned} \quad (49)$$

(2) and satisfying the boundary conditions

$$\begin{aligned} R_{ab}(t_0) &= R_0, \\ \omega_{ab}^b(t_0) &= \omega_0, \\ R_{ab}(t_f) &= R_f, \\ \omega_{ab}^b(t_f) &= \omega_f, \end{aligned} \quad (50)$$

where R_0 , ω_0 , R_f , and ω_f denote the initial angular configuration, the initial angular velocity, the terminal angular configuration, and the terminal angular velocity, respectively.

4.2. Transcription of the OAC on $SO(3)$ into an NLP Problem Based on LG-GPC Discretization. Different from the LG-PC direct transcription of an NLP problem in the Euclidean space [25], we transcribe the continuous Problem 9 into a discrete NLP problem by LG-GPC discretization. In general, an NLP problem has a standard form as follows.

Minimize the cost

$$f(z) \quad (51)$$

satisfying the equality and the inequality as follows:

$$\begin{aligned} z_{\min} &\leq z \leq z_{\max}, \\ c_E(z) &= 0, \\ c_{l,\min} &\leq c_l(z) \leq c_{l,\max}, \end{aligned} \quad (52)$$

where the vector $z \in \mathbb{R}^{n_x}$ is the whole decision variable of the NLP problem, $c_E(z)$ denotes the whole equality constraints, and $c_l(z)$ denotes the whole inequality constraints. Denote

$$c(z) := \begin{bmatrix} c_E(z) \\ c_l(z) \end{bmatrix} \quad (53)$$

with the constraints vector of the NLP problem.

For this purpose, we refer to the LG-PC transcription process introduced in [25], for transcribing Problem 9 into an NLP problem. Thus, the OAC problem on $SO(3)$ could be formulated as follows.

Algorithm 10 (transcription of the OAC problem on $SO(3)$ based on LG-GPC discretization).

Step 1. Mapping of the decision variables.

The total vector of optimization variables $z \in \mathbb{R}^{n_x(N+2)+n_uN}$ is given as

$$z = [z_x, z_u]^T, \quad (54)$$

where $z_x \in \mathbb{R}^{n_x}$ ($n_x = 6$) is given as

$$z_x = \begin{bmatrix} \chi_1 \\ \vdots \\ \chi_{n_x} \end{bmatrix}, \quad (55)$$

where the column vectors χ_j are given as

$$\chi_j = \begin{bmatrix} \chi_{j,1} \\ \vdots \\ \chi_{j,(N+2)} \end{bmatrix}, \quad j = 1, \dots, n_x, \quad (56)$$

where $\chi_{j,k}$ ($j = 1, \dots, n_x; k = 1, \dots, N+2$) is the value of the j th component of states at the k th discretization point.

Similarly, $z_u \in \mathbb{R}^{n_u}$ ($n_u = m$) is given as

$$z_u = \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_{n_u} \end{bmatrix}, \quad (57)$$

where the column vectors σ_j are given as

$$\sigma_j = \begin{bmatrix} u_{j,1} \\ \vdots \\ u_{j,N} \end{bmatrix}, \quad j = 1, \dots, n_u, \quad (58)$$

where $u_{j,k}$ ($j = 1, \dots, n_u; k = 1, \dots, N$) is the value of the j th component of the control at the k th LG collocation point.

Step 2. Construction of the constraint vector $c(z)$.

Step 2.1. Construction of the defect constraints $c_E(z)$.

For ω_{ab}^b , refer to (12) and (13), and define the defect constraints δ_k ($k = 1, \dots, N+1$) as

$$\begin{aligned} \delta_k &= \sum_{i=0}^N d_{ki} \omega_{ab}^b(\tau_i) \\ &\quad - \frac{t_f - t_0}{2} F(\omega_{ab}^b(\tau_k), R_{ab}(\tau_k), T_{\text{ext}}(\tau_k)), \\ &\quad k = 1, \dots, N, \end{aligned} \quad (59)$$

$$\delta_{N+1} = \omega_{ab}^b(t_f) - \omega_{ab}^b(t_0) - \sum_{i=0}^N \sum_{k=1}^N \bar{\omega}_k d_{ki} \omega_{ab}^b(\tau_i).$$

FIGURE 3: A schematic of a 3D rigid pendulum.

For configuration R_{ab} , refer to (26) and (27), and define the other part of the defect constraints δ'_k ($k = 1, \dots, N+1$) as

$$\begin{aligned} u(\tau_i) &= \text{cay}^{-1}(R_{ab}(t_k)^{-1} R_{ab}(\tau_i)), \quad i = 1, \dots, N, \\ \delta'_k &= \sum_{i=0}^N d_{ki} u(\tau_i) - \frac{t_f - t_0}{2} \tilde{G}(\omega_{ab}^b(\tau_k), u(\tau_k)), \\ &\quad k = 1, \dots, N, \end{aligned} \quad (60)$$

$$\delta'_{N+1} = u(t_f) - u(t_0) - \sum_{i=0}^N \sum_{k=1}^N \bar{\omega}_k d_{ki} \omega_{ab}^b(\tau_i).$$

Then we can define the matrix Δ as

$$\Delta = \begin{bmatrix} \delta_1^T & \delta_1'^T \\ \vdots & \vdots \\ \delta_{N+1}^T & \delta_{N+1}'^T \end{bmatrix}_{(N+1) \times 6}. \quad (61)$$

The matrix Δ can be then isomorphically mapped to the column vector of equality constraints using a pseudocoded version of the MATLAB column-stacking operator “:” as $\Delta(:) = c_E(z)$.

Step 2.2. Construction of the constraint vector $c(z)$, without inequality constraints.

We can represent the column vector $c(z)$ as

$$c(z) = [c_E(z)]. \quad (62)$$

Step 3. Automatic scaling of the NLP.

Step 4. Conversion of the NLP sparsity pattern.

Step 5. Separation of constraint and nonconstraint derivatives in the constraint Jacobian.

Step 6. Relation of the Lagrange multiplies of NLP with the costate of the continuous-time optimal control problem using costate mapping.

Step 7. Since solving the NLP from the geometric pseudo-spectral method results in controls at only the interior points, it is necessary to obtain the endpoint controls after the NLP is solved. The method for endpoint control computation is to employ the Pontryagin minimum principle using the endpoint values of the state and costate.

Remark 11. Since Steps 3~7 are more or less the same as those of [25], the repetitious details need not be given here.

4.3. Simulation Example. In this section, we take controlling an underactuated 3D pendulum as an example to examine the performance of the previous method. The 3D pendulum (illustrated in Figure 3) is, simply, a rigid body, supported at a fixed pivot point, which has three rotational degrees of freedom; it is acted on by a uniform gravity force and,

perhaps, by control and disturbance forces and moments [26].

The attitude kinematics and dynamics of the pendulum on SO(3) are given by

$$\begin{aligned}\dot{R}_{ab}(t) &= R_{ab}(t) \cdot \hat{\omega}_{ab}^b(t), \\ \dot{\omega}_{ab}^b(t) &= J^{-1} [J\omega_{ab}^b(t) \times \omega_{ab}^b(t) + m g \rho \times R_{ab}^T(t) e_3 \\ &\quad + Bu(t)],\end{aligned}\quad (63)$$

where the attitude configuration $R_{ab}(t)$ is a generalized coordinate of the rotation angular configuration of the body-fixed frame $\{O - X'Y'Z'\}$ relative to the inertial frame $\{O - XYZ\}$, satisfying

$$R_{ab} = \begin{bmatrix} \cos \theta \cos \psi & -\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi & \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi \\ \cos \theta \sin \psi & \cos \phi \cos \psi + \sin \phi \cos \theta \sin \psi & -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix}, \quad (64)$$

where ϕ, θ, ψ denote the Euler angles and the angular velocity vector $\omega_{ab}^b(t)$ in the body-fixed frame could be represented as $[p, q, r]^T$, in which p, q, r denote the angular velocity component, respectively, about X -, Y -, and Z -body axes. Thus, $\hat{\omega}_{ab}^b(t)$ is equal to $[0, -r, q, r, 0, -p; -q, p, 0]$. The conservative force $R_{ab}^*(t) \partial_{R_{ab}(t)} V(R_{ab}(t))$ could be easily obtained as $m g \rho \times R_{ab}^T(t) e_3$, where m is the mass of the pendulum and ρ is body-fixed vector from the pivot to the center of mass of the pendulum. $e_3 = [0, 0, 1]^T$ denotes the direction vector of gravity in the inertial coordinate frame. The nonconservative force $T_{\text{ext}}(t)$ satisfies $Bu(t)$ with B being the constant conjugate vector and the control vector $u(t)$ could be represented as $[u_1, u_2]^T$, in which u_1, u_2 denote the control components about X - and Y -body axes, respectively. The detailed properties of the 3D pendulum are presented in Table 2.

Consider the OAC problem of the 3D pendulum on SO(3), which is to control the pendulum from the initial fixed angle $(\phi_0, \theta_0, \psi_0)$ and the initial fixed angular velocity (p_0, q_0, r_0) to the terminal fixed angle $(\phi_f, \theta_f, \psi_f)$ and the terminal fixed angular velocity (p_f, q_f, r_f) with the minimum moment of the external force. The simulation parameters of the OAC problem are presented in Table 3.

The proposed optimal control method (LG-GPC) is also compared with the prior works in [7, 9–11], which are also based on geometrically exact computations (Lie group variational integrator) on SO(3). There are two major categories of methods for solving the discrete first-order optimal necessary conditions derived from the optimal control problem: the nonlinear root-finding method (NRF) and the shooting method. In terms of the former, Bloch et al. [9] and Hussein et al. [10] propose a geometric structure-preserving optimal control method on SO(3), which applies a standard Newton

iterative approach to solve the discrete first-order necessary optimality conditions. However, this method does not consider the moment of the conservative force and therefore could not be compared with our method immediately. To this end, we resort to a more general discrete geometric optimal control method on Lie groups proposed in [11], which consider both the moment of conservative force and that of external nonconservative force. In this method, for the 3D pendulum, the dimension of discrete decision variables is $3(2N + 1)$ with the discretization number N being 15 and the maximum iterative number being $2e^4$. Termination tolerances on the constraint violation, the function value, and argument are $1e^{-6}$, $1e^{-4}$, and $1e^{-6}$, respectively. In terms of the latter, Lee et al. [7] use a line search with the backtracking algorithm, referred to as “Newton-Armijo iteration (NAI),” to solve the discrete first-order necessary optimality conditions, where the time stepsize is $h = 0.001$ s and the number of integration steps is $N = 1000$. Figure 4 gives the optimal states and controls obtained from the LG-GPC method when compared with LG-PC and NRF.

As seen from Figure 4, the results obtained from LG-GPC and LG-PC satisfy the boundary condition of states and control, except those of NRF, because the first two could consider constraints of states and control explicitly, the latter has no choice but to introduce a dummy factor. Table 4 illustrates the optimal costs $\mathcal{J}$ and operation time t of LG-GPC, LG-PC, NRF, and NAI. It is concluded from the optimal costs that the methods based on LGVI are superior to LG-GPC and LG-PC, since the former belong to indirect method that is higher in accuracy. However, the last two have not only larger convergence radius, but also higher convergence rate. With the numerical advantage of the geometric collocation discretization, LG-GPC has the least operation time.

— LG-GPC
- - LG-PC

(a) Angular configuration versus time

... NRF

(b) Angular configuration versus time

— LG-GPC
- - LG-PC

(c) Angular velocity versus time

... NRF

(d) Angular velocity versus time

— LG-GPC
- - LG-PC

(e) Control versus time

... NRF

(f) Control versus time

FIGURE 4: The optimal state and control of the pendulum obtained by LG-GPC, LG-PC, and NRF.

TABLE 2: Properties of a 3D rigid pendulum [7].

Physical quantity	Symbol	Value
Mass	m	1 kg
Moment of inertia	J	$\begin{bmatrix} 0.156 & 0 & 0 \\ 0 & 0.156 & 0 \\ 0 & 0 & 0.3 \end{bmatrix} \text{ kg}\cdot\text{m}^2$
Vector of pivot	ρ	$[0, 0, 0.75]^T \text{ m}$
Conjugate vector	B	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}^T$
Weighting matrix	P	$I_{2 \times 2}$

TABLE 3: The simulation parameters of the OAC problem.

Decision variable	Initial value	Terminal value	Boundary condition	Initial guess
ϕ (rad)	0	π	$[-\pi, \pi]$	0
θ (rad)	0	0	$[-\pi, \pi]$	0
ψ (rad)	0	$\pi/2$	$[-\pi, \pi]$	0
p (rad/s)	0	0	$[-\pi, \pi]$	0
q (rad/s)	0	0	$[-\pi, \pi]$	0
r (rad/s)	0	0	$[-\pi, \pi]$	0
u_1 (N·m)	—	—	$[-5, 5]$	0
u_2 (N·m)	—	—	$[-5, 5]$	0

TABLE 4: Optimized performance index and operation time.

	LG-GPC	LG-PC	NRF	NAI
$\mathcal{J}$	11.3466	10.9628	2.1044	1.52
t	4.3838	4.8782	154.7895	—

5. Conclusions

This paper shows how to apply the pseudo-spectral collocation method to a constrained OAC problem for a rigid body on SO(3). First, we establish a left-invariant rigid-body attitude dynamical model in a body-fixed frame on SO(3). Then, we propose a geometric collocation method which has a comprehensive advantage in accuracy, computational efficiency, and the Lie group structural conservativeness. Preservation of key motion invariance leads to robust dynamics approximation. Finally, the proposed method is applied to a constrained OAC problem.

For future work, we will further compare the method with more typical numerical methods in the Euclidean space and on the Lie group, to test the efficiency of our method. Moreover, we will extend the proposed method to a broader category of left-invariant rigid-body dynamics systems in engineering.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (61403410).

References

- [1] A. V. Rao, "A survey of numerical methods for optimal control," in *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, AAS Paper 09-334, AAS/AIAA, Pittsburgh, Pa, USA, August 2009.
- [2] F. Lewis and V. Syrmos, *Optimal Control*, John Wiley & Sons, 1995.
- [3] L. S. Pontryagin, V. Boltyanskii, R. Gamkrelidze, and E. Mischenko, *The Mathematical Theory of Optimal Processes*, Interscience Publishers, New York, NY, USA, 1962.
- [4] H. B. Keller, *Numerical Solution of Two Point Boundary Value Problems*, SIAM, 1976.
- [5] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, vol. 12 of *Texts in Applied Mathematics*, Springer, New York, NY, USA, 2002.
- [6] F. Bullo and R. Murray, "Proportional derivative (PD) control on the Euclidean group," in *Proceedings of the European Control Conference (ECC '95)*, Rome, Italy, September 1995.
- [7] T. Lee, M. Leok, and N. H. McClamroch, "Optimal attitude control of a rigid body using geometrically exact computations on SO(3)," *Journal of Dynamical and Control Systems*, vol. 14, no. 4, pp. 465–487, 2008.
- [8] T. Lee, *Computational Geometric Mechanics and Control of Rigid Bodies*, ProQuest, Ann Arbor, Mich, USA, 2008.
- [9] A. M. Bloch, I. I. Hussein, M. Leok, and A. K. Sanyal, "Geometric structure-preserving optimal control of a rigid body," *Journal of Dynamical and Control Systems*, vol. 15, no. 3, pp. 307–330, 2009.
- [10] I. I. Hussein, M. Leok, A. K. Sanyal, and A. M. Bloch, "A discrete variational integrator for optimal control problems on SO(3)," in *Proceedings of the 45th IEEE Conference on Decision & Control*, pp. 6636–6641, IEEE, San Diego, Calif, USA, December 2006.
- [11] M. B. Kobilarov and J. E. Marsden, "Discrete geometric optimal control on lie groups," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 641–655, 2011.
- [12] R. Moulla, L. Lefèvre, and B. Maschke, "Geometric pseudospectral method for spatial integration of dynamical systems," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 17, no. 1, pp. 85–104, 2011.
- [13] K. Engø, "On the construction of geometric integrators in the RKMK class," Tech. Rep. 158, Department of Computer Science, University of Bergen, Bergen, Norway, 1998.
- [14] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, 1994.
- [15] D. A. Benson, *A gauss pseudospectral transcription for optimal control [Ph.D. thesis]*, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Mass, USA, 2004.
- [16] H. Munthe-Kaas, "Runge-kutta methods on Lie group," *BIT*, vol. 35, pp. 572–587, 1995.
- [17] J. S. Hesthaven, S. Gottlieb, and D. Gottlieb, *Spectral Methods for Time-Dependent Problems*, Cambridge University Press, 2007.
- [18] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

- [19] T. Lee, N. H. McClamroch, and M. Leok, "A lie group variational integrator for the attitude dynamics of a rigid body with applications to the 3D pendulum," in *Proceedings of the IEEE Conference on Control Applications (CCA '05)*, pp. 962–967, IEEE, Toronto, Canada, August 2005.
- [20] A. Iserles, H. Z. Munthe-Kaas, S. P. Nørsett, and A. Zanna, "Lie-group methods," *Acta Numerica*, vol. 9, pp. 215–365, 2000.
- [21] A. Iserles and S. P. Nørsett, "On the solution of linear differential equations in Lie groups," *Philosophical Transactions of the Royal Society A*, vol. 357, no. 1754, pp. 983–1020, 1999.
- [22] L. N. Trefethen, *Spectral Methods in Matlab*, SIAM, Philadelphia, Pa, USA, 2000.
- [23] J. R. Dormand and P. J. Prince, "A family of embedded Runge-Kutta formulae," *Journal of Computational and Applied Mathematics*, vol. 6, no. 1, pp. 19–26, 1980.
- [24] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, vol. 31 of *Springer Series in Computational Mathematics*, Springer, Berlin, Germany, 2002.
- [25] A. V. Rao, D. A. Benson, C. L. Darby et al., "Algorithm 902: GPOPS, a MATLAB software for solving multiple-phase optimal control problems using the gauss pseudospectral method," *ACM Transactions on Mathematical Software*, vol. 37, no. 2, article 22, 2010.
- [26] J. Shen, A. K. Sanyal, N. A. Chaturvedi, D. Bernstein, and N. H. McClamroch, "Dynamics and control of a 3D pendulum," in *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC '04)*, pp. 323–328, Paradise Island, Bahamas, December 2004.